This exam is to be done individually. It is due, in paper form, **STAPLED**, at the start of class on Wednesday, March 24, 2003. The exam is worth 100 points. 11 pages, 14 problems.

You may fill out the exam either electronically or by hand (or a combination of the two). In either case, **PRINT** the exam and turn in a paper copy. If you fill out the exam electronically, do not change the page breaks. If you need more space, you may attach additional pages.

You may use any reference you like: books, notes, web pages, etc. However, the exam you turn in must be your own work; you may not discuss solutions with other people in the class.

1. [5 pts] Described how a transformation stack (as in OpenGL) makes it convenient to draw hierarchical objects.

2. [4 pts] What is “Constructive Solid Geometry”, and how is it used in CG?
3. [10 pts total] **BSP Trees.**
   
   3a. [2 pts] List two graphical effects/algorithms that BSP trees are useful for.

   
   3b. [3 pts] Using BSP trees can be problematic with some types of scenes. Explain.

   
   3c. [5 pts] Draw two triangles below, label the edges with the numbers 1 through 6, and draw a diagram of the BSP tree for these edges, assuming all normal vectors point outward.
4 [5 pts total] **Jittering.**
   4a. [2 pts] What does “jittering” mean in CG?

   4b. [3 pts] Give two effects that can be produced using jittering, and briefly explain how jittering would be used to produce each effect.

5. [3 pts] List some of the pro’s & con’s of using Catmull-Rom splines vs. using Bézier curves & surfaces.
6 [8 pts total] **Affine Transformations.**

6a. [2 pts] What exactly is an “affine transformation”?

6b. [2 pts] Which of the transformations we have studied are affine? (*You might find it simpler to say which are not.*)

6c. [4 pts] When we deal with curve & surface generation, we usually want our curves & surfaces to behave in certain ways when affine transformations are applied to them. Explain.
7. [8 pts total] **Representing Rotations.**
   
   7a. [6 pts] List three ways of representing rotations. For each give an advantage or disadvantage that it has over the other methods.

7b. [2 pts] OpenGL represents rotations in two ways. What are they? *(Hint: What are the two ways you can tell OpenGL to do a rotation?)*
8. [10 pts total] **The Painter’s Algorithm.**
   8a. [2 pts] What is the “Painter’s Algorithm”?

8b. [2 pts] List two effects that the P.A. is useful for producing.

8c. [3 pts] The P.A. needs a preprocessing step. What are two ways this step can be performed?

8d. [3 pts] List some trade-offs of using the P.A. vs. using the Reverse P.A.
9. [8 pts total] **Scene Graphs.**
   
   9a. [2 pts] One way to represent a scene is with a scene graph. What sorts of operations are made convenient by this representation?

   9b. [3 pts] Briefly describe the trade-offs of storing a scene graph as a tree structure vs. storing it as a DAG.

   9c. [3 pts] Consider a robot with a body and two legs, each of which has a movable foot. Draw a reasonable scene-graph representation for this robot.
10. [6 pts total] **Shadow Volumes.**
   10a. [3 pts] Briefly describe the “shadow-volumes” technique for producing shadows.

10b. [3 pts] List some guidelines for when the S.V. method is a good one to use.

11. [5 pts total] **Polynomial Curves.** Suppose I want to use a polynomial to describe a curve. I want my curve to satisfy a list of constraints.
   11a. [2 pts] How do I typically determine what degree the polynomial should be?

11b. [3 pts] How do I typically determine the coefficients of the polynomial?
12. [10 pts total] **HSR Methods.**
   12a. [2 pts] Explain “object-space” vs. “image-space” HSR.

   12b. [3 pts] Many traditional HSR methods (z-buffering, etc.) are rather inefficient when used with large, complex scenes. What needs to be done to improve efficiency for such scenes?

   12c. [5 pts] Briefly describe a method that is efficient for some types of large, complex scenes.
13. [8 pts total] **VR Programming.**
   13a. [4 pts] List some of the major services that VR libraries typically provide to application programs.

   13b. [4 pts] List some ways in which a program written for a VR environment will differ significantly from a program written for a traditional desktop environment. *(Think in terms of program organization and interactions with libraries and hardware. I am not interested so much in user-interface design issues here; save those for the problem 14.)*
14. [10 pts total] **VR User Interface.** Many user-interface components/ideas cannot be moved unmodified from a desktop-style program to a VR program.

14a. [2 pts] Name one such user-interface component.

14b. [3 pts] Describe why desktop-style implementations of this can be inappropriate in VR.

14c. [5 pts] Describe a solution to this problem. How would you design/handle/implement this user-interface component in a VR environment?